Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего

образования «Южно-Уральский государственный университет (национальный

исследовательский университет)»

Высшая школа экономики и управления

Кафедра Информационных технологий в экономике

Программирование на языке Python (курс молодого бойца)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ

по дисциплине «ПРОГРАММИРОВАНИЕ»

Рецензент, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Руководитель, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020г. «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020г.

Нормоконтролёр, \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Автор, студент группы ЭУ-120

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ Плесовских Н.С./\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020г.

Работа защищена с оценкой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020г.

Челябинск 2020

Челябинск 2020

# Аннотация

Плесовских Н. С. – Челябинск: ЮУрГУ, ЭУ - 120, 2020. - 10с.

Данная курсовая работа состоит из трех частей.

В курсовой работе были написаны и исследованы на эффективность 3 основные метода сортировки (bubble, insert, select). Рассмотрение библиотеки Numpy и решение 31 задачи. Также в третьей части рассмотрен метод решения «СЛАУ» с помощью метода Гаусса и библиотеки numpy.

Аннотация. 2

Введение. 3

Курсовая работа — часть 1: "Методы сортировки". 5

1.1. Сортировка одномерного массива. 5

1.1.1. Сортировка включением (insert). 5

1.1.2. Обменная сортировка (bubble). 6

1.1.3. Сортировка выбором (select). 7

1.2. Анализ эффективности. 8

Курсовая работа — часть 2: Матричная математика и работа с пакетами, библиотекой Nampy. 9

Курсовая работа — часть 3: Решения СЛАУ методом Гаусса на PYTHON . 10

Заключение. 11

# Введение

Python является языком программированием которой стремительно набирает свою популярность и широко масштабность в контингенте.

В данной работе рассмотрено способы применения python для решения определенных, поставленных задач. Так же способ использования языка программирования с дополнительными пакетами и решения типовых задач «СЛАУ» методом Гаусса.

С помощью разнообразных библиотек, python позволяет выполнять и решать множество, различных задач которые очень важны как для начинающего, так и для опытного программиста.

Цели работы:

1. С помощью практического применения и методом анализа выявить наиболее эффективный метод сортировки.
2. С помощью библиотеки numpy решить 31 матрицу.
3. С помощью numpy решить «СЛАУ» методом Гаусса.

# Курсовая работа — часть 1: "Методы сортировки"

## Алгоритмы сортировки.

Сортировка есть процесс перестановки множества объектов в определенном порядке. Суть сортировки — это облегчить поиск нужных элементов и отсеивание остального «песка». Сортировка используется во многих деятельностях повсеместно.

В данной работе я написал три различных кода, для трех различных методов сортировок. Массив состоял из 10000 символов, которые сортировались по порядку. Что дало понять, что скорость зависит как от вычислительной способности процессора, так и от метода сортировки.

### Сортировка включением (insert).

Сортировка вставками (англ. Insertion sort) — алгоритм сортировки, в котором элементы входной последовательности просматриваются по одному, и каждый новый поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов.

На вход алгоритма подаётся последовательность n чисел. Сортируемые числа также называют ключами. Входная последовательность на практике представляется в виде массива с n элементами. На выходе алгоритм должен вернуть перестановку исходной последовательности

В начальный момент отсортированная последовательность пуста. На каждом шаге алгоритма выбирается один из элементов входных данных и помещается на нужную позицию в уже отсортированной последовательности до тех пор, пока набор входных данных не будет исчерпан. В любой момент времени в отсортированной последовательности элементы удовлетворяют требованиям к выходным данным алгоритма.

Данный алгоритм можно ускорить при помощи использования бинарного поиска для нахождения места текущему элементу в отсортированной части. Проблема с долгим сдвигом массива вправо решается при помощи смены указателей.

### Обменная сортировка (bubble)

Этот простой алгоритм выполняет итерации по списку, сравнивая элементы попарно и меняя их местами, пока более крупные элементы не «всплывут» в начало списка, а более мелкие не останутся на «дне».

Сначала сравниваются первые два элемента списка. Если первый элемент больше, они меняются местами. Если они уже в нужном порядке, оставляем их как есть. Затем переходим к следующей паре элементов, сравниваем их значения и меняем местами при необходимости. Этот процесс продолжается до последней пары элементов в списке.

При достижении конца списка процесс повторяется заново для каждого элемента. Это крайне неэффективно, если в массиве нужно сделать, например, только один обмен. Алгоритм повторяется n² раз, даже если список уже отсортирован.

### Сортировка выбором (select)

При сортировке массива Arr0, Arr2, ..., ArrN‑1 методом простого выбора среди всех элементов находится элемент с наименьшим значением Arri, и Arr0 и Arri обмениваются значениями. Затем этот процесс повторяется для получаемого подмассива Arr1, Arr2, ..., ArrN‑1, ... Arrj, Arrj+1, ..., ArrN‑1 до тех пор, пока мы не дойдем до подмассива ArrN‑1, содержащего к этому моменту наибольшее значение.

Для метода сортировки простым выбором оценка требуемого числа сравнений – N(N‑1)/2. Порядок требуемого числа пересылок, которые требуются для выбора минимального элемента, в худшем случае составляет O(N2). Однако порядок среднего числа пересылок есть O(N\*Lg(N)), что в ряде случаев делает этот метод предпочтительным.

## Анализ эффективности.

При помощи программы определяем эффективность трех алгоритмов методов сортировки (вставками (insertion), обменная (bubble), выбором (selection)).

Исходя из полученный данных можем сделать вывод, что наиболее эффективным методом сортировки является сортировка выбором (select).  
Все итоги, данные и код можно найти на [github](https://github.com/NikitaPlesovskix/cource-work.py)([1](https://github.com/NikitaPlesovskix/cource-work.py/tree/master/First%20part)).

# часть 2: Матричная математика и работа с пакетами.

Сам по себе python служит только для несложных вычислений. Особенность python — это его расширяемость. Поскольку экосистема python огромна, существуют бесчисленные библиотеки, упрощающие программирование на этом языке. Благодаря таким библиотекам становится возможным более легкое выполнение определенных задач без необходимости написания излишнего кода.

Одной из самых значимых библиотек для превращения python в математический пакет являются numpy.

numpy - это библиотека языка python, которая предоставляет общие математические и числовые операции в виде пре-скомпилированных, быстрых функций. То есть, numpy предоставляет базовые методы для манипуляции с большими массивами и матрицами. Для того, чтобы понять, как работает данная библиотека, я решил 31 задачу с помощью nampy, решение которых предоставляю в свободно доступе на [github](https://github.com/NikitaPlesovskix/cource-work.py)([2](https://github.com/NikitaPlesovskix/cource-work.py/tree/master/Second%20part)).

# часть 3: Решение слау

Метод Гаусса — классический метод решения системы линейных алгебраических уравнений (СЛАУ). Назван в честь немецкого математика Карла Фридриха Гаусса. Это метод последовательного исключения переменных, когда с помощью элементарных преобразований система уравнений приводится к равносильной системе треугольного вида, из которой последовательно, начиная с последних (по номеру), находятся все переменные системы. В основе метода лежит идея последовательного исключения неизвестных с помощью подстановок, суть которой состоит в приведении данной системы к другой, равносильной ей, но более простой системе. Это приведение одной системы к другой осуществляется путем элементарных преобразований, которые производятся над уравнениями системы или, что удобнее, над строками расширенной матрицы. Элементарному преобразованию системы линейных уравнений соответствует одноименное элементарное преобразование строк ее расширенной матрицы.   
 Разобравшись, что представляет из себя метод Гаусса, я решил СЛАУ методом Гаусса с помощью пакета numpy. Всего у меня было пять матриц, которые я записал в таблице Excel, после чего сохранил ее на рабочий стол в формате CSV. Что же такое CSV? Это текстовый файл, в котором содержится определенная информация, в моем случае это матрицы, а сами поля разделяются специальными символами, или как их еще называют – разделителями. После этого, я перенес данный файл в PyCharm, переименовал его в “numpy-gauss” и, с помощью команды “CTRL+R”, заменил разделитель «;» на «,». Далее я начал работать над кодом решения СЛАУ методом Гаусса.  
Код, «СЛАУ» и результат можно найти на [github](https://github.com/NikitaPlesovskix/cource-work.py)([3](https://github.com/NikitaPlesovskix/cource-work.py/tree/master/Third%20part)).

# Заключение.

Благодаря простоте и гибкости языка python, его можно рекомендовать пользователям (математикам, физикам, экономистам и т.д.) не являющимся программистами, но использующими вычислительную технику и программирование в своей работе.

Исходя из всей информации можно сделать вывод, python это язык программирования общего назначения, нацеленный в первую очередь на повышение продуктивности самого программиста, нежели кода, который он пишет. Говоря простым человеческим языком, на python можно написать практически что угодно (веб-/настольные приложения, игры, скрипты по автоматизации, комплексные системы расчёта, системы управления жизнеобеспечением и многое другое) без ощутимых проблем. Но для того что бы полностью раскрыть потенциал python, нужно использовать дополнительные библиотеки.